UNIVERSITE IBA DER THIAM de THIES
(U.I.D.T)

Établissement -UFR des Sciences Économiques et Sociales-UFR des Sciences et Technologie
Département Commun - Management des Organisations - Mathématiques - Informatique
Sciences Économiques et de Gestion

PROJET D'ALGÈBRE LINÉAIRE

MBERY SENE GNING
MASTER 1 SCIENCE DES DONNEES ET APPLICATIONS:
OPTION IIA

15 octobre 2024

Exercice 1

Introduction à l'Algèbre Linéaire

L'algèbre linéaire est une branche fondamentale des mathématiques qui étudie les systèmes d'équations linéaires, les vecteurs, les matrices et les espaces vectoriels. Elle constitue un outil essentiel pour analyser des données, modéliser des systèmes complexes et résoudre une multitude de problèmes.

L'importance de l'algèbre linéaire se manifeste dans sa capacité à traiter des problèmes multidimensionnels de manière systématique. Elle est largement utilisée dans des domaines tels que la statistique, pour l'analyse de données multivariées et la construction de modèles prédictifs ; en ingénierie, pour concevoir et analyser des systèmes dynamiques ; et en économie, pour modéliser les interactions entre divers agents économiques.

Dans ce projet, nous utiliserons le logiciel R, un environnement de programmation et un langage de statistiques qui facilite l'analyse de données et la réalisation de calculs mathématiques complexes. R est particulièrement prisé pour sa capacité à manipuler des matrices et à effectuer des opérations algébriques, ce qui le rend idéal pour nos travaux en algèbre linéaire.

Définitions des Termes Clés de notre projet

Matrice : Une matrice A de n * p éléments ; est un tableau de nombres à n lignes et p colonnes. Une telle matrice est dite de dimension (n,p) ;de format (n,p) ou même d'ordre (n,p) : Format =(nombre de lignes, nombre de colonnes)

Matrice Diagonale : Une matrice où tous les éléments en dehors de la diagonale principale sont nuls. Les matrices diagonales simplifient les calculs, notamment lors de la multiplication ou de l'inversion.

Transposée : La matrice transposée d'une matrice A est obtenue en échangeant ses lignes et ses colonnes. Cette opération est cruciale pour les produits scalaires et les transformations linéaires.

Déterminant : Une valeur scalaire calculée à partir d'une matrice carrée, fournissant des informations sur ses propriétés, comme son inversibilité. Un déterminant non nul indique qu'une matrice est inversible.

Valeurs Propres et Vecteurs Propres : Pour une matrice A, une valeur propre λ est un scalaire tel que $A\mathbf{v} = \lambda\mathbf{v}$ pour un vecteur non nul $\mathbf{v}$. Ces concepts sont fondamentaux pour l'analyse des systèmes dynamiques.

Matrice de Passage : Une matrice qui permet de transformer des vecteurs d'une base à une autre. Elle est utilisée pour passer d'une représentation vectorielle à une autre, facilitant l'analyse dans différents systèmes de coordonnées.

Matrice Inverse : Une matrice A^{-1} qui, multipliée par la matrice originale A, produit la matrice identité. L'inversion de matrices est essentielle pour résoudre des systèmes d'équations linéaires.

En résumé, l'algèbre linéaire est un outil mathématique puissant, dont les applications sont vastes et variées. À travers ce projet, nous explorerons ces concepts à l'aide de R, renforçant notre compréhension et notre capacité à appliquer ces notions à des problèmes pratiques.

Exercice 2

Soit $A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -1 \end{bmatrix}$

Question 1 Déterminer la matrice transposée.

```
#question 1
A <- matrix(c(2, 0, 1, 1, 1, 1, -2, 0, -1), nrow = 3, byrow = TRUE)
A_transpose <- t(A)
A_transpose
```

```
> #question 1
> A <- matrix(c(2, 0, 1, 1, 1, 1, -2, 0, -1), nrow=3, byrow=TRUE)
>
> A_transpose <- t(A)
> A_transpose
     [,1] [,2] [,3]
[1,]    2    1   -2
[2,]    0    1    0
[3,]    1    1   -1
```

Question 2 Calculer le produit de la matrice A et de sa transposée.

```
produit <- A %*% A_transpose
produit
```

```
> produit <- A %*% A_transpose
> produit
     [,1] [,2] [,3]
[1,]    5    3   -5
[2,]    3    3   -3
[3,]   -5   -3    5
>
```

Question 3 Calculer le déterminant de A.

```
det_A <- det(A)
det_A
```

```
> det_A <- det(A)
> det_A
[1] 0
>
```

Question 4 Calculer les valeurs propres et les vecteurs propres de A.

```
eigen_A <- eigen(A)
valeurs_propres <- eigen_A$values
vecteurs_propres <- eigen_A$vectors
valeurs_propres
vecteurs_propres
```

```
> eigen_A <- eigen(A)
> valeurs_propres <- eigen_A$values
> vecteurs_propres <- eigen_A$vectors
> valeurs_propres
[1] 1 1 0
> vecteurs_propres
     [,1]       [,2]       [,3]
[1,]    0  0.7071068 -0.4082483
[2,]    1  0.0000000 -0.4082483
[3,]    0 -0.7071068  0.8164966
>
```

Question 5 Déterminer la matrice diagonale.

```
D <- diag(valeurs_propres)
```

```
D
```

```
> D <- diag(valeurs_propres)
> D
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    0
>
```

Question 6 Former la matrice de passage notée P. Calculer l'inverse de la matrice de passage notée P^{-1}.

```
P <- vecteurs_propres
P_inv <- solve(P)
P
P_inv
```

```
> P <- vecteurs_propres
> P_inv <- solve(P)
> P
     [,1]       [,2]       [,3]
[1,]    0  0.7071068 -0.4082483
[2,]    1  0.0000000 -0.4082483
[3,]    0 -0.7071068  0.8164966
> P_inv
         [,1] [,2]     [,3]
[1,] 1.000000    1 1.000000
[2,] 2.828427    0 1.414214
[3,] 2.449490    0 2.449490
>
```

Question 7 Effectuer le produit matricielle entre P et P^{-1}.

```
identite <- P %*% P_inv
identite
```

```
> identite <- P %*% P_inv
> identite
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
```

```
[3,]    0    0    1
>
```

Exercice 3

Question 1 A partir de ces données, former une matrice 5*5(5 lignes et 5 colonnes).

```
# Definir le vecteur original
data <- c(2, 4, 3, 1, 3, 7, 2, 4, 1, 3,
          3, 7, 4, 5, 4, 4, 5, 5, 4, 4,
          1, 5, 3, 4, 2, 5, 4, 6, 3, 2,
          2, 2, 4, 3, 4, 6, 3, 2, 5, 4,
          4, 4, 5, 5, 3, 4, 6, 4, 6, 3)

# Creer une matrice 5x5
M <- matrix(data[1:25], nrow = 5, byrow = TRUE)

# Afficher la matrice
M
```

```
 # Definir le vecteur original
> data <- c(2, 4, 3, 1, 3, 7, 2, 4, 1, 3,
+           3, 7, 4, 5, 4, 4, 5, 5, 4, 4,
+           1, 5, 3, 4, 2, 5, 4, 6, 3, 2,
+           2, 2, 4, 3, 4, 6, 3, 2, 5, 4,
+           4, 4, 5, 5, 3, 4, 6, 4, 6, 3)
> # Creer une matrice 5x5
> M <- matrix(data[1:25], nrow = 5, byrow = TRUE)
> # Afficher la matrice
> M
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    3    1    3
[2,]    7    2    4    1    3
[3,]    3    7    4    5    4
[4,]    4    5    5    4    4
[5,]    1    5    3    4    2
>
```

Question 2 Calculer le determinant de la matrice.

```r
#determinant de la matrice
d=det(M)
d
```

```
> #determinant de la matrice
> d=det(M)
> d
[1] -93
>
```

Question 3 : Déterminer la matrice transposée.

```r
M_transpose=t(M)
M_transpose
```

```
> M_transpose=t(M)
> M_transpose
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    7    3    4    1
[2,]    4    2    7    5    5
[3,]    3    4    4    5    3
[4,]    1    1    5    4    4
[5,]    3    3    4    4    2
```

Question 4 : Calculer la matrice inverse.

```r
M_inverse <- solve(M)
M_inverse
```

```
> M_inverse <- solve(M)
> M_inverse
            [,1]        [,2]        [,3]       [,4]        [,5]
[1,] -0.12903226  0.27956989  0.19354839 -0.3333333  0.05376344
[2,]  0.29032258  0.20430108  0.06451613 -0.6666667  0.46236559
[3,]  0.22580645  0.01075269 -0.83870968  0.3333333  0.65591398
[4,] -0.48387097 -0.11827957  0.22580645  0.3333333 -0.21505376
[5,] -0.03225806 -0.43010753  0.54838710  0.6666667 -1.23655914
>
```

Question 5 : Extraire l'élément qui se trouve à la 4e ligne et 3e colonne.

```r
element <- M[4, 3]
print(element)
```

```
> # Extraire l'element a la 4e ligne et 3e colonne
> element <- M[4, 3]
> print(element)
[1] 5
>
```

Question 6 : Afficher la 3e ligne.

```r
# Afficher la 3e ligne de la matrice
troisieme_ligne <- M[3, ]
print(troisieme_ligne)
```

```
> # Afficher la 3e ligne de la matrice
> troisieme_ligne <- M[3, ]
> print(troisieme_ligne)
[1] 3 7 4 5 4
>
```

Question 7 : Afficher la 4e colonne.

```r
quatrieme_colonne <- M[ ,4]
print(quatrieme_colonne)
```

```
> quatrieme_colonne <- M_5x5[ ,4]
> print(quatrieme_colonne)
[1] 1 1 5 4 4
>
```

Question 8 : Former une matrice M(5 * 4)et une matrice N(4 * 8) enfin de calculer le produit M * N.

```r

# Creer une matrice M de dimensions 5x4
M <- matrix(data[1:20], nrow = 5, ncol = 4)
M

# Creer une matrice N de dimensions 4x8
N <- matrix(data[1:32], nrow = 4, ncol = 8)
N
```

```

# Calculer le produit M * N
produit <- M %*% N

# Afficher le resultat
print(produit)
```

```
> # Creer une matrice M de dimensions 5x4
> M <- matrix(data[1:20], nrow = 5, ncol = 4)
> M
     [,1] [,2] [,3] [,4]
[1,]    2    7    3    4
[2,]    4    2    7    5
[3,]    3    4    4    5
[4,]    1    1    5    4
[5,]    3    3    4    4
> # Creer une matrice N de dimensions 4x8
> N <- matrix(data[1:32], nrow = 4, ncol = 8)
> N
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    2    3    1    4    5    1    2    3
[2,]    4    7    3    5    5    5    5    2
[3,]    3    2    3    4    4    3    4    2
[4,]    1    4    7    4    4    4    6    2
> # Calculer le produit M * N
> produit <- M %*% N
> # Afficher le resultat
> print(produit)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]   45   77   60   71   73   62   75   34
[2,]   42   60   66   74   78   55   76   40
[3,]   39   65   62   68   71   55   72   35
[4,]   25   36   47   45   46   37   51   23
[5,]   34   54   52   59   62   46   61   31
```